



Azure™ Services Platform



Cloud and Connectivity

Customers need the flexibility to run their applications and services on a variety of platforms: on-premises, in the cloud, and everywhere in between. In heterogeneous environments, customers need three things to be successful:

- they need to connect and compose applications,
- they need to do it easily through consumable web services, and
- they need to maintain interoperability across platforms, protocols, and standards.

We have made Microsoft .NET Services available to fulfill these three customer needs. It is designed to connect platforms of any kind – across the cloud and cloud-to-on-premises.

Flexible, Internet-scale Service Bus

Microsoft .NET Services is a set of loosely coupled services that make it easy to realize many of the benefits offered by cloud-based software. These services are particularly useful for integration scenarios—by using .NET Services, you can develop federated cloud-based applications without having to entirely depend on your internal infrastructure team. It enables you to make your applications more scalable and flexible, resulting in more rapid deployments while

keeping your local infrastructure intact.

Standards-based & Interoperable

While you will appreciate the similarity and completeness of these hosted services with the Microsoft platform and Microsoft tools, they also fully support industry standards and Web protocols such as WS-* standards, representational state transfer (REST), SOAP, and HTTP. As a result, .NET Services can be accessed from non-.NET platforms.

Extend Existing Investments

Although cloud services is a relatively a new area for most developers, you can leverage your existing .NET skills. .NET Services is well suited for developers with .NET and Visual Studio® development experience. It does not require developers to learn a new set of tools. Common tasks can be performed through a simple Web user interface, and advanced programmatic access is available through the .NET Services software development kit (SDK).

Microsoft .NET Services

While more services are in the works, .NET Services is currently comprised of these core services.

Access Control

The Access Control Service takes the pain out of securing your Web applications and services. A simple

and familiar programming model keeps your code clean and lets you move a lot of the difficult to implement functionality to a declarative model of rules and claims. These rules can be easily configured to cover all your current and future security needs.

Want to offer your application to a corporation as a hosted/cloud-based solution, but the customer is reluctant to create and manage thousands of user accounts on your site? With the Access Control Service, you can easily federate in your customer's existing identity management infrastructure, be it Active Directory®, Tivoli Directory Server, or any other standards-based infrastructure. All you need to do is configure the Access Control Service, and then your application just works. Your customer does not need to create and manage parallel user accounts on your Web site and will be much more receptive to a hosted/cloud-based solution.

Need to integrate your application with a new identity system you have never heard of? Chances are the Access Control Service already supports it, or you can easily extend it to bring the new identity system into your world without any changes to your application.

The service is based on a claims-based authorization model, which alleviates the need to understand a



variety of identity providers and their architecture.

In addition, the service's federation capabilities allow use of: -

- Active Directory accounts.
- Windows Live™ ID.
- X509 certificates.
- User name and password.
- Managed cards/personal cards.

Service Bus

The Service Bus alleviates the pain that you have when you try to expose your application's service functionality when it is located behind a network address translation (NAT) and/or bound to frequently changing, dynamically assigned IP addresses.

If you have a problem of creating global hierarchical namespace, Service Bus lets you create one that is also DNS-independent and transport-independent. In addition, your service can also get a stable Internet-accessible Uniform Resource Identifier (URI), irrespective of location.

If you want to simplify the interactions between services, use the messaging service in the .NET

Services Service Bus, which together with the identity and workflow service, acts as an "Internet Service Bus"—this allows your services to publish and subscribe in a loosely coupled manner.

Additionally, depending on your security needs, Service Bus can also be secured by the Access Control Service. This could automate authorization rules enforcement and dramatically simplify access and control management of your application service.

Workflow Service

If you want to declaratively configure a predefined set of simple activities, the .NET Services Workflow Service is in the cloud for you. Workflow Service works as an agent to orchestrate your service interactions, allowing you execute your set of simple activities. It includes activities that can send, receive, and inspect XML messages as well as .NET activities to control flow.

It is easy to design workflows with a simple drag-and-drop operation from predefined activities into Visual

Studio designer. It is relatively simple to manage with the .NET Service hosted tools and APIs—by using them, you can deploy, manage, and track the execution of workflow instances running on the service.

The service is designed and architected for cloud scenarios with transparent scalability of underlying components (for example, persistence stores) and distribution of load between hosts. On-premises applications could also benefit by having some activities offloaded to be executed by a scalable cloud based workflow.

Resources and Links

For more information, visit:

<http://www.azure.com> and <http://msdn.microsoft.com/azure>

Download the SDK and the Client Redistributable for .NET Services at: <http://www.azure.com/netservices>

Get started today:

<http://www.azure.com/register.msp>

Follow .NET Services at:

<http://twitter.com/dotnetservices>

Azure Services Platform

Microsoft
SQL Services

Microsoft
.NET Services

Live Services

Microsoft
SharePoint
Services

Microsoft
Dynamics CRM
Services

Windows® Azure™